International Standard “OGC® Moving Features” to address “4Vs” on locational BigData

Akinori Asahara
and Hideki Hayashi
Hitachi Ltd.
Center for Technical Innovation – Systems,
Resaerch and Development Group
Kokubunji-shi, Tokyo, Japan
Email: {akinori.asahara.bq,
hideki.hayashi.xu}@hitachi.com

Nobuhiro Ishimaru
Hitachi Ltd.,
Defense Systems Company
Chiyoda-ku, Tokyo, Japan
Email: nobuhiro.ishimaru.yu@hitachi.com

Ryosuke Shibasaki
and Hiroshi Kanasugi
Center for Spatial Information Science
University of Tokyo
Meguro-ku, Tokyo, Japan
Email: shiba@csis.u-tokyo.ac.jp
yok@iis.u-tokyo.ac.jp

***Abstract*—Applications utilizing many types of location data, such as traffic congestion estimation and facility management using indoor pedestrian tracks, have been rapidly increasing. Such applications require the integration of various locational data from different data sources to produce more values. Efforts to ensure smoother data exchange are required for promoting the use of such applications because handling and integrating location data will enlarge the market for geo-spatial information. In response to this need, we had proposed a data encoding standard called ‘OGC® Moving Features’ to contribute to smoother data exchange, and it was adopted as an international standard on Feb. 2015. We demonstrate in this work that OGC® Moving Features is an effective tool to advance technologies for applications using many types of location data, with referring “4Vs” to represent the most pressing bigdata issues.**

I. Introduction

The location data of a mobile object is a typical type of big data handled by a geospatial information system (GIS). Large amounts of location data are produced by positioning systems such as the GPS (Global Positioning System). Not only that, the quantity of the location data being produced is rapidly increasing day by day. A representative examples is the widespread use of personal cellar phones. Most people these days have cell phones that connect to the Internet by communicating with wireless bases. The records of such communication, known as call detail records (CDRs), are available for determining the locations of the cell phones. Moreover, many kinds of devices have function as actual GPS receivers, such as vehicle navigation systems and automatic identification systems (AISs) of maritime and aviations vessels. Additionally, cameras and radar systems track moving objects, and the resulting trajectories are also location data. Accordingly, a huge amount of location data (called “locational bigdata” hereafter) is being produced and stored.

Therefore, the demand for handling locational bigdata is very rapidly increasing. There are numerous applications using locational bigdata that are being considered and implemented, including traffic congestion information services using probe cars equipped with GPS, the tracking of automatic trucks for logistics management, and agent-based road traffic simulation systems for forecasting traffic situations. Moreover, the growth of smartphones in the market has created an enormous market for geospatial applications that require the integration of locational bigdata from many heterogeneous data sources.

Technologies to address the issues involved in handling locational bigdata are thus surely needed. First, we need to ask oursevels, what exactly are the issues? We can use ‘4Vs’ to represent the most pressing bigdata issues: volume, variety, velocity, and value. How exactly are these issues at play when it comes to locational bigdata? It is not clear.

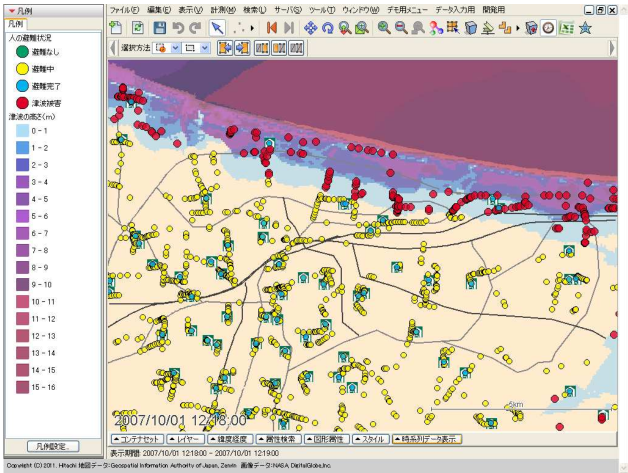
In response to the above, we have proposed a new international standard “OGC® Moving Features” to the international standardization organization Open Geospatial Consortium (OGC) [1]. The OGC® Moving Features provides encoding rules of trajectories, so it is applicable for locational bigdata. In this paper, we summarize how the OGC® Moving Features standard contributes to solving the “4Vs” problems. Also, because the “velocity” issue is not solvable with only OGC® Moving Features, a combination with the MQTT protocol is proposed. In this paper, we clarify which issues have been resolved and which still need to be addressed. This will be of benefit to the researchers working on developments in the field.

II. Related works

”Trajectory analysis”, which is a technology to extract information from stored trajectory data, is a representative technology for locational bigdata. Many types of trajectory analyses exist, such as trajectory smoothing [2] [3] for reducing errors in positioning, trajectory clustering [4] [5] [6] for retrieving similar trajectories from a database, an algorithm for labeling positioning data [7], and the extraction of a “representative path”, which is abstract information about a trajectory dataset [8] [9].

Movement prediction is an important useage of trajectory analysis. Statistical models used for the prediction describe a probability formula that expresses the transitions between positions. Various Markov-chain models [10]–[13] have been used to improve prediction accuracy.

Another application is density estimation of moving objects suce as pedestrians. The density of pedestrians in a shopping



©Hitachi, Ltd.

Fig. 1. Tsunami simulation integrated with evacuation simulator. Yellow circles represent locations of evaluating people, and red circles represent victims of the tsunami.

mall, for example, is understood as the degree of crowdedness at each shop. Kernel-density estimators [14], [15] are popular methods for estimating the density of points.

As shown above, many technologies have already been developed. However, these technologies require various functions to handle locational bigdata, namely, a common data encoding format, spatio-temporal database platforms, a data collection infrastructure and so on. Sometimes such basic technology is absent, and for promoting focusthe development of such technologies, the basis should be prepared. Thus, the basis using international standards is our in this paper.

III. "VALUE": WHY LOCATIONAL BIGDATA

As stated in the previous section, many applications for locational bigdata are being considered and implemented. For such applications, it makes more sence from a business value standpoint to combine the locational bigdata with another geospatial data. For example, the location data of an object exhibits only the location itself, but by combining it with another piece of geospatial data, we can ge a more detailed idea of the situation around the object. Many vehicles stopped on a road, for example, suggests that there may have been a traffic accident on the road. Thus, systems using single-source location data will evolve into more integrated systems. The following are examples of such efforts.

1) Integrated simulation for disaster risk management: Different simulation systems such as people evacuation simulations, road vehicle simulations including emergency vehicles, and tsunami simulations, should be integrated. Figure 1 shows an example of a tsunami simulation and a people evacuation simulation integrated on a GIS platform. Many of these systems (excepting the tsunami simulations) are agent-based simulation systems that output locational bigdata describing an individual agent's tracks.

2) Security services: Security management officers such as police officers need services to estimate criminal risk. This service requires sharing of situational information as a common picture by integrating and visualizing data on pedestrian and vehicle movements collected from heterogeneous sensors such as surveillance cameras, GPSs, and mobile phones.

3) Meteorology: Tracks of harricanes and typhoons are stored into spatio-temporal databases. Using datamining algorithms, their courses are predicted. The tracks can produce other informative estimation with integration to other data. For example, the number of people afftected by typhoons are estimated with integrating population distribution.

4) Traffic information services: Traffic congestion and the trafficability of roads can be estimated from real-time vehicle trajectory data collected from vehicles. The information is provided as guidance information to road users and maintainers of roads. The data sources are diverse, coming from a fleet of taxis with GPS, trucks, buses, and navigation system users. It is threfore more necessary to integrate data encoded in different ways. In one real-life example, the trajectory data from car navigation devices were integrated to determine the trafficability of road segments after the Great Earthquake of East Japan in 2011 [16].

5) Navigation for Robots: Control technologies for various robots and autonomous vehicles have recently been rapidly improved. These robots have to move around while avoiding collision with other objects. Location data of the robots and the people around them are collected, making this data a form of locational bigdata. Since robots can identify only nearby obstacles and moving objects with laser range and vision sensors, they may require situational information on a larger scale, and this requires integration of the trajectory data collected through sensor networks.

6) Maritime vessels: Positioning devices are installed on maritime vessels to ensure the safety and security within governmental maritime sectors. Some of these devices are mandatorily requested by regional and international standardization initiatives (e.g. EU Directives). For the maritime use cases, the following information has to be included at least:

1) ship position provided by different data sources (e.g. AIS)
2) voyages that describe the tracks of vessels

Additional information about the vessel incidents and the ship particulars is considered complementary.

7) Aviation: Many tracks of aircraft and other airborne vessels are used by governmental aviation sectors. The main sources of such data are surveillance radar measuring position and heading of the aricraft. In addition, an active response from the aircraft transponders supplies additional information such as its identity. Several data-encoding standards for thisntype of data have been established, but they are applicable for only aviations.

8) Indoor tracks: Laser imaging detection and Ranging (LiDAR) is a technique useful for detecting and tracking pedestrians [17], [18]. Pedestrian-tracking systems using LiDAR are often used to investigate the tendencies of pedestrians movement in big facilities such as shopping malls and train terminals. Such trajectory data are recorded on a spatio-temporal database and the population distribution of the pedestrian tracks can be calculated by a trajectory analysis system. The

TABLE I. REPRSENTATIVE POSITIONING SYSTEMS

System	Indoor support	Type	ID
GPS	no	continuous	consistent
IR sensor	yes	cell-ID-based	consistent
RSS	yes	continuous	consistent
Camera	yes	continuous	consistent during capturing
LiDAR	yes	continuous	consistent during capturing

population can be calculated by counting the number of people in each grid. From another viewpoint, the locations where many pedestrians are likely to cluster, potentially creating a bottleneck, can be predicted by counting the number of pedestrians around each booth. Such clusters need to be reduced, and the counts are useful for cluster reduction.

9) Sports: Sports fileds are another key point for locational bigdata application. As an example, a soccer game use case [19] is described here. Soccer coaches typically analyze past matches to improve their tactics and prepare for the next match and the tracks of players and the ball may produce very helpful information for such tactics planning. During a soccer match, 22 players and one ball are in constant movement around the soccer field, their tracks are obtained with video analysis and sensors such as RFIDs. The statistics of the trajectories of the ball and players show that depends on the tactics. For example, defensive tactics can be better understood by analyzing the shape of the defensive players rather than the position of each individual defender separately. On the basis of such information, more information will be found by extracting primitive behaviors such as ball possession, dribbles, intercepts, and passes. Such information is useful for planning complicated strategies such as formations.

IV. "VARIETY": OGC® MOVING FEATURES ENCODING STANDARD

A. Data sources

A track consisting locational bigdata is defined as a time series of position data. The position data consists of several bits of information, namely, an ID (to identify the moving object), measurement time, and measured coordinate values. Positioning systems like GPS (Global Positioning System) provide such position data. There are many additional kinds of positioning systems are available for the applications mentioned above.

Table I lists a selection of such positioning systems. Although GPS is the most popular, it is not suitable for indoor use because it uses signals from satellites, so indoor positioning systems have been developed [10]. Two of the most famous indoor positioning systems are radio frequency ID (RFID) tags [20], [21] and infrared (IR) sensors [22]. This type of positioning system, called the "cell-ID-based positioning", provides only the IDs of locations. In contrast, positioning system like GPS give continuous coordinate values (e.g. longitude and latitude). Generally, continuous coordinate values include more information than discrete ones, so continuous coordinate values are suitable for extracting knowledge. Some positioning systems use radio signals but also provide continuous coordinates by using radio signal strength (RSS) [23] [24] [25]. Because location IDs of locations can be

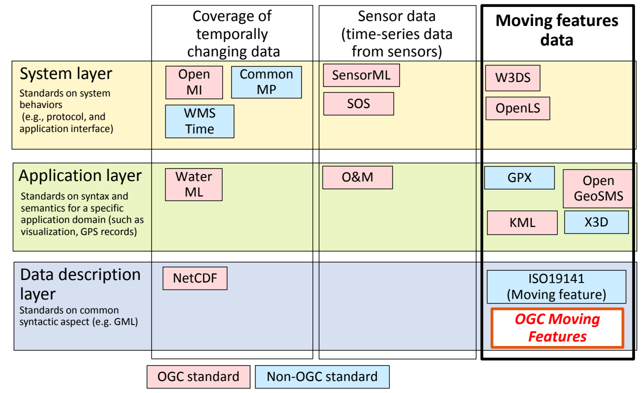


Fig. 2. Standards on spatio-temporal data. OGC® Moving Features is a basic standard on encoding tracks of moving objects.

converted into coordinate values, trajectory description using coordinate values are applicable for both cases.

Video analysis and LiDAR-based systems also provide track data. Over the last few years, several pedestrian-tracking systems using LiDAR have been developed [17], [18], providing precise (errors less than 30 cm) pedestrian-track data. Moreover, many video-analysis technologies [26], [27] also extract tracks from color images captured by video camera. A depth camera, which captures images including depth information, is similarly applicable to the detection of moving objects [28], and accuracy with a depth camera is higher than that with a video camera. The tracks obtained by LiDAR and videos can also be described with coordinate values (x, y, z). The coordinate systems are different from longitude and latitude. Moreover, moving objects identification is sometimes difficult for LiDAR and videos. For example, a person captured by LiDAR has ID to represent the parson. However, once the person become out of the frame of the LiDAR, the ID is changed because LiDAR does not identify who is the person. Thus, ID of trajectory itself is needed, which is independant of the moving object ID.

A simulation is also considered as another type of data sources. As shown above, people or vehicles are often simulated to estimate the numer of them, such as evacuation simlation. Such track data produces more value with integrating other data (e.g., tsunami simulation). Data description in a simulation softerware is sometimes very compact (for example, locations of people are identified by IDs of roads). However, the data should be converted into coordinates to integrate other data because such specific encoding is not compatible with the other system. Therefore, trajectory description with coordinate values is reasonable even for simulation data.

Accordingly, tracks are generally described with a series of coordinate values. However, many kinds of coordinate systems might be used (e.g. indoor, outdoor). Locational bigdata thus requires a standard and flexible (i.e. able to support many standard coordinate systems) encoding format for trajectory data.

B. Requirements for standard and OGC® Moving Features

One of the most well-known standard organizations, the International Standard Organization (ISO), recommends the

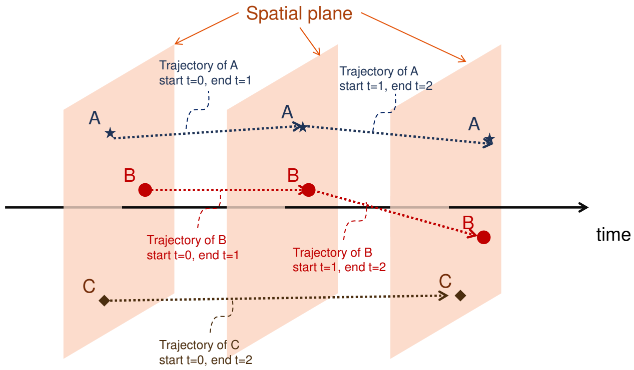


Fig. 3. Foliation model applied to OGC® Moving Features. ISO19141:2008 supports the model to describe trajectories of moving objects.

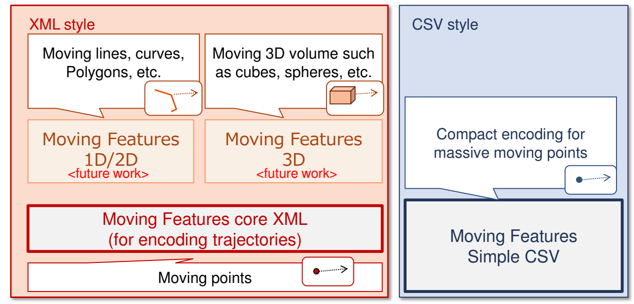


Fig. 4. OGC® Moving Features modularity. Two OGC® Moving Features specifications are adopted: XML style and CSV style. Both regurates encoding of point-like trajectories.

"ISO19141:2008 geographic information – Schema for moving features" [29] as an abstract data model. Although this model provides a basic model to describe trajectory data, the implementation schema, such as XML schema, JSON schema, and so on, are not defined. Figure 2 summarizes the standards related to temporally changing geospatial data. Many standards for coverage data and sensor observation data have already been established, and the visualization of moving objects is already supported by popular standards such as X3D [30] and KML [31]. However, the adopted standard for moving features was only ISO19141:2008, which is an abstract standard. From a practical point of view, there is a strong need for an implementation standard that can facilitate actual data exchange.

For this reason, we had proposed a new implementation standard for encoding trajectory data, called "OGC® Moving Features", and it was adopted by OGC in Feb. 2015 [32]. OGC® Moving Features was developed as an implementation specification of ISO19141:2008 with satisfying following requirements:

1) "Schema for Moving Features (ISO19141, 2008)" should be referred to as the conceptual framework.
2) A standard data model should describe the movement of zero to three-dimensional geometric features including changes in attitude or rotation along with the movement.
3) The implementation specifications should be prioritized.
4) Unnecessary overlaps should be avoided, while popular standards should be referred to in the development of a new specification on the moving features.

Figure 3 illustrates the data model used in OGC® Moving Features. The model is called the "foliation model", which is inherited from ISO19141:2008. Trajectories of three moving points (A, B and C) are shown in the figure. The horizontal axis indicates time, and three planes represent spatial coordinates as a temporal snapshot. Each trajectory, which connects two temporal snapshots, has a start time and an end time. At t=0 (start of all data), A and C start moving, and B stays. Then, at t =1, the movement of A is changed, and B starts moving. In this case, the trajectory of A from t=0 to t=1, the trajectory of A from t=1 to t=2, the trajectory of B from t=0 to t=1, the trajectory of B from t=1 to t=2, and the trajectory of C from t=0 to t=2 are encoded. Note that the trajectory of which speed is zero (e.g., that of B from t=0 to t=1) is recorded nevertheless. That is, changes of state, including location, moving velocity, and attributes, are encoded. The encoded trajectories are ordered by time. It enable to determine the states of all features even if only the first half of the data is loaded.

Figure 4 shows the modularity of the OGC® Moving Features standard. Moving Features core XML is the most fundamental specification with high extensibility. OGC® Moving Features XML defines an XML element to encode tracks of point-like features. Moving Features Simple CSV is another style of encoding. This was defined to reduce data size even if a massive amount of data is encoded. The other specifications to support shapes of features, such as Moving Features 1D/2D and Moving Features 3D (shown in the figure), will be defined as extensions of Moving Features core XML in future. This demonstrates the extensibility of Moving Features core XML in comparison to Moving Features Simple CSV.

An example of XML data describing trajectories is shown in Fig. 5. Moving Features core XML is defined as a GML [33] application and definition written with XML schema is provided. Thus general GML parsers are applicable to parsing the Moving Features core XML data. Moreover, a new element inheriting the Moving Features XML element can be easily defined.

`mf:LinearTrack` element, inheriting `gml:feature`, is generally used for tracks that can be interpolated with linear functions. Every mf:LinearTrack describes a short part of a trajectory. `mfIdRef` attribute of `mf:LinearTrack` is an identifier of a moving object which the data describes, and `start` and `end` attributes indicate time range of the trajectory's existance. By collecting `mf:LinearTrack` elements which have common `mfIdRef` attribute and ordering by `start` attribute, the entire track can be obtained. Furthermore, the spatial reference system is stated in `mf:sTBoundedBy` element, namely, `srsName` attribute of `gml:EnvelopeWithTimePeriod` element, defined in GML, is inherited by `mf:LinearTrack`. Because GML is one of the most popular and general encoding schema for geospatial objects, most coordinate systems thus are supported.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<mf:MovingFeatures xmlns:mf="http://schemas.opengis.net/mf-core/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.opengis.net/mf-core/1.0 moving_features_gml_app.xsd"
  gml:id="MFC_0001">
  <mf:sTBoundedBy offset="sec">
    <gml:EnvelopeWithTimePeriod srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
      <gml:lowerCorner>0.0 0.0</gml:lowerCorner>
      <gml:upperCorner>5.0 5.0</gml:upperCorner>
      <gml:beginPosition>2012-01-17T12:33:41Z</gml:beginPosition>
      <gml:endPosition>2012-01-17T14:00:00Z</gml:endPosition>
    </gml:EnvelopeWithTimePeriod>
  </mf:sTBoundedBy>
  <mf:header>
    <mf:VaryingAttrDefs>
      <mf:attrDef>
        <mf:attrDef name="state" type="xsd:integer" />
        <mf:attrDef name="type" type="xsd:integer" />
      </mf:attrDef>
    </mf:VaryingAttrDefs>
  </mf:header>
  <mf:foliation>
    <mf:LinearTrajectory gml:id="LT001" mfIdRef="a" start="0" end="1050">
      <gml:posList>1.0 1.0 2.0 3.0</gml:posList>
      <mf:Attr>1001,12</mf:Attr>
    </mf:LinearTrajectory>
    <mf:LinearTrajectory gml:id="LT002" mfIdRef="b" start="0" end="2400">
      <gml:posList>1.0 2.0 1.0 3.0</gml:posList>
      <mf:Attr>1001,2</mf:Attr>
    </mf:LinearTrajectory>
    <mf:LinearTrajectory gml:id="LT003" mfIdRef="a" start="1050" end="2410">
      <gml:posList>2.0 3.0 1.0 1.0</gml:posList>
      <mf:Attr>1002,12</mf:Attr>
    </mf:LinearTrajectory>
    <mf:LinearTrajectory gml:id="LT004" mfIdRef="b" start="2400" end="5000">
      <gml:posList>1.0 2.0 1.0 3.0</gml:posList>
      <mf:Attr>1001,2</mf:Attr>
    </mf:LinearTrajectory>
    <mf:LinearTrajectory gml:id="LT005" mfIdRef="a" start="2410" end="5000">
      <gml:posList>2.0 3.0 1.0 1.0</gml:posList>
      <mf:Attr>1001,2</mf:Attr>
    </mf:LinearTrajectory>
  </mf:foliation>
</mf:MovingFeatures>
```

Fig. 5. An example of OGC® Moving Features XML. All elements used in this example are defined in the OGC® Moving Features XML Core specification.

V. "VOLUME": SIMPLE ENCODING STYLE CSV

The huge data size of locational bigdata might be an issue for some applications. For example, imagine the tracks of one million people in a large city being collected through smart-phones every minute. Because one day consists of 1,440 minutes, one billion pieces data will be recorded in this scenario. If one location data size is 1 kbyte, the data size of one day reaches 1TB and that of several years will be 1 PB. Such a huge amount of data is difficult to handle, and a simple encoding style to reduce data size is thus strongly needed.

As mentioned in the previous section, OGC® Moving Features Simple CSV, which is a data size-friendly encoding style, was developed for thiis purpose, that is, as a compact data encoding. Figure 6 shows an example of data description using Moving Features Simple CSV. A piece of data encoded by Moving Features CSV has two main parts: a header part and a body part. The header part, in which the line starts with '`@`', provides boundary and attribute definitions, and the body part, which follows the header, is used to describe the trajectories themselves. In the body part, every track is encoded as a line, as '`a,0,1050,1.0 1.0 2.0 3.0,1001,12`.' The ID of a moving object is shown in the first column, the start time and end time of the track follow, the shape of the track is shown in the next, and the attributes are shown as the last columns. The shape is expressed as a line string.

This encoding style has fewer flexibility and extensibility than the XML style, but the data size is very small. Because the data size of encoded Moving Features depends mainly on the number of trajectories, the data size can be roughly estimated using the data size of each trajectory. Namely, one line data size of CSV text shown in Fig. 6 is around 40 bytes. In addition, if the locations of one million people in a city are collected through cell phones every minute, 1,440,000,000 tracks (that is, 24 hours $\times$ 60 minutes $\times$ 1,000,000 people) are stored [34] (this setting is similar to exisisting dataset [34]). The data size is 57.6GB in that case.

For comparison, the data size with OGC® Moving Features XML is estimated. OGC® Moving Features XML encodes a track as follows:

```xml
<mf:LinearTrajectory gml:id="a1" mfIdRef="a"
  start="0" end="1050">
 <gml:posList>1.0 1.0 2.0 3.0</gml:posList>
 <mf:Attr>1001,2</mf:Attr>
</mf:LinearTrajectory>
```

The data size of this code is around 160 bytes, which is four times of that encoded by CSV. Thus, the entire data size is 230.4 GB. Another possible encoding style is JSON, which is compatible with Java script. For instance, GeoJSON [35] is

```
@stboundedby,urn:x-ogc:def:crs:EPSG:6.6:4326,,0.0␣0.0,5.0␣5.0,2012-01-17T12:33:41Z,2012-01-17T14:00:00Z,sec
@columns,mfidref,trajectory,typecode,xsd:integer,mode,xsd:integer
a,0,1050,1.0␣1.0␣2.0␣3.0,1001,12
b,0,2400,1.0␣2.0␣1.0␣3.0,1001,2
a,1050,2410,2.0␣3.0␣1.0␣1.0,1002,12
b,2400,5000,1.0␣2.0␣1.0␣3.0,1001,2
a,2410,5000,1.0␣2.0␣1.0␣3.0,1001,2
```

Fig. 6. An example of OGC® Moving Features CSV. Two header lines which start with '@' are shown. Many trajectory lines follow them.

TABLE II. DATASIZE ESTIMATION

Style	Datasize	Required parser	Advantage
CSV text	57.6GB	CSV Parser	Easy to read
XML	230.4GB	XML parser	Extendable metadata
GeoJSON	259.2GB	Java Script parser	Web friendly
Binary	83.5GB	Binary parser	high performance

the most popular JSON schema describing geometric objects. Using GeoJSON, one linear track is encoded as follows.

```
{"type": "Feature",
 "geometry": {"type": "LinerTrajectory",
 "coordinates": [[1.0, 1.0], [2.0 3.0]]},
 "properties'':{"mfidref":"a",  "starttime":"0",
   "endtime":"1050", "attr": ["1001","2"] },
```

The data size is around 180 bytes. Thus the entire data size is 259.2GB. Additionally, the most compact encoding style is encoding as a binary data stream. All data to be encoded are as follows: the ID of a moving object, the start time and end time of the track, the shape of the track, and the attributes. The ID takes 2 bytes including a delimiter if the ID is "a". The data size of start and end time is 16 bytes because time is encoded with 8 bytes integer. Trajectory shape requires 32 bytes because four coordinate values are encoded with 8bytes double precision floating point number. The attributes "1001" and "2" can be encoded with 4 bytes integer, so 8 bytes are required. Accordingly, one trajectory data size is 58 bytes in total, and entire datra size is 83.52 GB. This is greater than that of CSV. A CSV line will be longer if a large dataset is encoded because text describing time and coordinate values become longer, however the data size with binary would be comparable with that with CSV nevertheless.

This rough estimation of data size is summarized in Table II. Data sizes by CSV text and binary are around 1/4 of that by XML and JSON. Implementaion of software handling XML and JSON data is easy because existing pasers are available for loading XML and JSON data. Binary encoding, that requires a new paser, has a drawback in this standpoint. CSV text also requires a new parser, but such paser is easy to develop because CSV is a common and simple format. Therefore, CSV encoding is adpted as a simple encoding style. Besides, binary encoding is being considered as one of the next steps because it has another advatages: perfomance of processing is quite high. Furthermore, JSON encoding is also positioned as future work because of easiness of implementation especially for web site.

VI. "VELOCITY" - FAST DATA EXCHANGE API

Ideally, locational big data should be collected in real time, as discussed above. High-throughput protocol to send data is therefore required. If one million people send their locations every five minutes, the required throughput is more than 3,000 record/sec. This throughput is very high, so for handling such requests using HTTP, very powerful and expensive hardware is required. To keep costs reasonable, we need a faster API to send locational big data in real time.

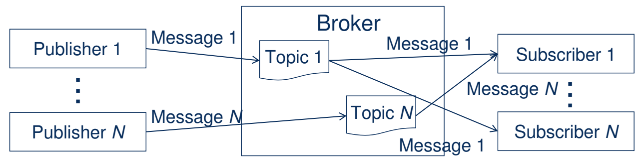


Fig. 7. Publisher-subscriber model used by MQTT. The broker recieves messages from multiple publishers, and distributes to multiple subscribers. Subscribers must register in advance to recieve messages. A topic represents a category of messages, and the broker distributes messages referring it's topic.

In most cases, an API should provide many functions; authentication, registration, metadata sharing, and so on. Several APIs with such functions are currenly in use, but they are implemented on HTTP, too slow. Therefore, non-HTTP APIs are needed.

We therefore propose an API using a stream protocol named message queuing telemetry transport (MQTT) [36] to send Moving Features Simple CSV data. MQTT uses a publish-subscribe message pattern to provide one-to-many message distribution, as shown in Fig. 7. The message sent from a publisher is registered as a topic, which is a named logical channel classifying messages into a hierarchical structure (e.g. `/sensor/1/temperature`, `/sensor/1/pressure`). The message registered as the topic is delivered to one or multiple subscribers that applied to that topic for a delivery.

Figure 8 shows the system architecture of an MQTT-based data-collection system. The components of this system are classified into three main parts: publishers, subscribers, and brokers. "Sensor" is an entity that provides data of an observed property as output. Multiple sensors make up a system called "sensor system". A "location data aggregator" receives location data from sensor systems and sends location data to location data receivers. These receivers (which are applications) receive the location data from location data aggregators.

Figure 9 shows a sequence for data collection by the proposed protocol. The processes in the sequence are classified into two steps: registration of sensors and applications ((1) in the figure), and sending location data ((2) in the figure).

The registration processes are defined for providing functions such as registration and metadata sharing. "`RegisterSensor`" is an operation for registering the information of a sensor system to a location data aggregator while the registration processes. The data aggregator deletes the information of a sensor system when it receives an empty `RegisterSensor` message from the sensor system.

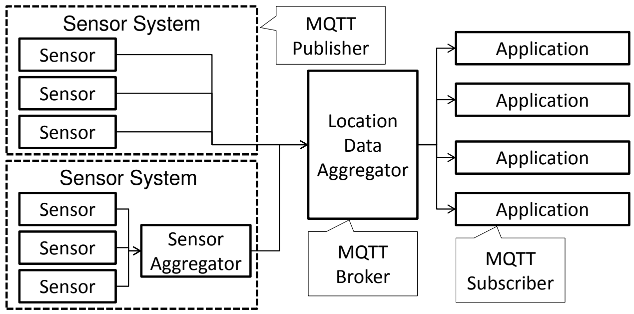


Fig. 8. Moving-Features-data-collection system. A set of sensors managed in a system is called a sensor system here. MQTT topic is defined for each sensor system to distribute location data to each application.

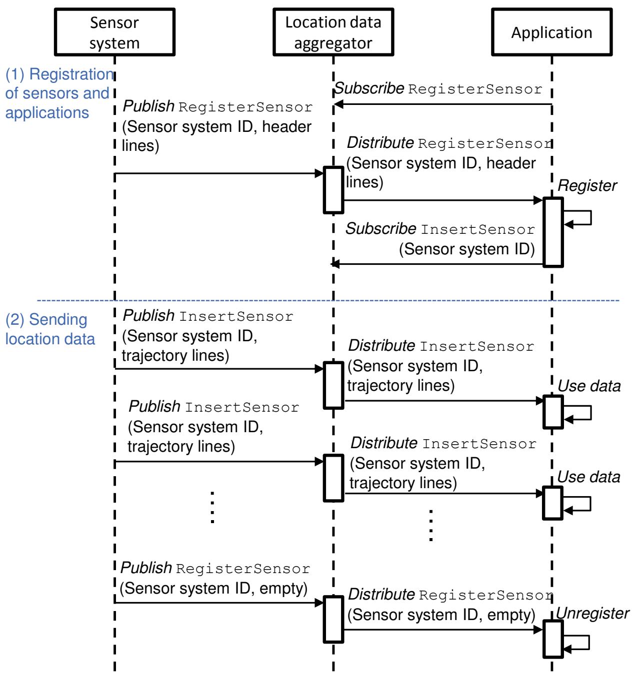


Fig. 9. The sequence of the protocol. It's processes are categorized into two types: "(1) registration of sensors and applications" and "(2) sending location data". Most functions such as initialization are carried out while the Registration.

"`InsertTrajectory`" is an operation for sending location data from sensor systems to applications. The applications requiring sensor data have to register as subscribers of `InsertTrajectory` in the registration processes.

The sending processes are processes to send location data periodically. The sensor systems publish `InsertTrajectory` messages including location data in their payload. The location data aggregator receives them, and it distributes them to the applications.

The detailed specification is presented in the following sections.

A. *Registration of sensors and applications*

The registration for a sensor system is as follows:

1) The sensor system sets a topic "`lcp/RegisterSensor/`[SensorSystemID]" and a schema definition (which is a header part defined by OGC® Moving Features Simple CSV), where "`lcp`" comes from acronyms of "location collection protocol".

The registration for an application is as follows.

1) The application sends a request for registration as a subscriber of topic "`lcp/RegisterSensor/#`" to the location data aggregator. "#" is a wild card in topic definition of MQTT.
2) The location data aggregator registers the application.
3) The location data aggregator sends the latest message of topic "`lcp/RegisterSensor/#`". This message includes sensor system IDs and schema definitions generated by the sensor systems.
4) The application sends a request for registration as a subscriber of topic "`lcp/InsertTrajectory/`[SensorSystemID]"
5) The location data aggregator registers the application.

B. *Sending location data*

Sending and receiving location data processes are as follows:

1) The registering sensor system sends location data encoded by OGC® Moving Features Simple CSV data body to the location data aggregator on topic "`lcp/InsertTrajectory/`[SensorSystemID]".
2) The location data aggregator sends the location data to applications which are subscribers of topic "`lcp/InsertTrajectory/`[SensorSystemID]".

When a sensor system stops sending location data, it unregisters with an empty `RegisterSensor` message as follows:

1) The sensor system sends a message with an empty payload to the location aggregator on topic "`lcp/RegisterSensor/`[SensorSystemID]".
2) The location data aggregator sends the message to applications registering as subscribers of "`lcp/RegisterSensor/`[SensorSystemID]".

C. *Experimental performance estimation*

To confirm the feasibility of this API, an experimental evaluation of throughput was conducted. For comparison, the most popular protocol HTTP was also evaluated.

Evaluation environments were as follows. Apache Tomcat was used as the HTTP Web server. The method to send location data was very simple: GET method with arguments that include (x, y). Mosquito and Paho Java were used as the MQTT environment. A parameter to determine quality of service (QoS) in MQTT was tested for all cases (QoS1, QoS2, and QoS3). Data size of one piece of location data was set to 100 bytes.

The evaluation was conducted as follows. First, 1,000 records/sec was set as the data-sending rate and the records that actually arrived were counted. We use the ratio of the arrival rate to data-sending rate as the criteria of this evaluation. A ratio less than 100 % indicates that the throughput is

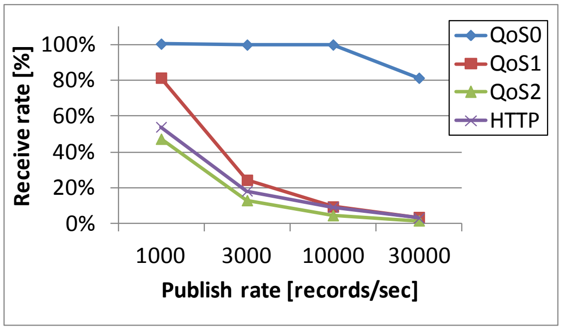


Fig. 10. Performance comparison. QoS 0 of MQTT achieved outstanding results.

not enough. Various data-sending rates were tested: 3,000 records/sec, 10,000 records/sec and 30,000 records/sec.

Figure10 plots the results of the experiment. Horizontal axis indicates the data-sending rate and vertical axis indicates the ratio to the actual throughput. Only MQTT QoS0 achieved 100% when the data-sending rate was less than 10,000 records/sec. The other setting never achieved 100% even if 1,000 records/sec was set as the data-sending rate. 1,000 records/sec equals a case in which 60,000 people send their location every seconds. This situation is not particularly unusual, and even so MQTT QoS1, MQTT QoS2, and HTTP were insufficient for such cases. This demonstrates that MQTT QoS0 is the only possible solution. MQTT QoS0 does not ensure that all data is distributed; that is, some of data might not arrive. However, in many cases, a lack of location data does not create a problem so long as the missing pieces of data are relatively few. Therefore, the API to send OGC® Moving Features CSV data using MQTT QoS0 is feasible for many applications.

VII. Conclusion

For locational bigdata, issues referred to as the "4Vs" need to be addressed in order to create more "value". OGC® Moving Features, which is a common and simple trajectory-data encoding standard, is helpful for exchanging "various" data without increasing data "volume". Furthermore, communication protocol using MQTT is useful for exchanging OGC® Moving Features data in order to accelerate "velocity" of data distribution. Accordingly, a clue for the 4Vs are provided by OGC® Moving Features. We consider that OGC® Moving Features is helpful for advancing technologies relating to locational bigdata.

Actually, there is another "V" – "Veracity", that is, data quality. Data quality issues are also in scope of the OGC® standard, and these shall be the focus of our future work.

Acknowledgment

A part of this work is supported by a consignment to research and development of techniques on the use and application of real-time information in the G-space platform from the Ministry of Internal Affairs and Communications, Japan. The authors greatly appreciate the kind and helpful supports.

References

[1] "Open Geospatial Consortium," http://www.opengeospatial.org/.

[2] F. Chazal, D. Chen, L. Guibas, X. Jiang, and C. Sommer, "Data-driven trajectory smoothing," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 251–260.

[3] M. van Kreveld and L. Wiratma, "Median trajectories using well-visited regions and shortest paths," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 241–250.

[4] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2007, pp. 593–604.

[5] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2005, pp. 491–502.

[6] F. Elias, G. Kostas, and T. Yannis, "Index-based most similar trajectory search," in *Proceedings of IEEE 23rd International Conference on Data Engineering*, 2007, pp. 816–825. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4221730

[7] Y. Asakura and E. Hato, "Tracking survey for individual travel behaviour using mobile communication instruments," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 3-4, pp. 273 – 291, 2004, intelligent Transport Systems: Emerging Technologies and Methods in Transportation and Traffic. [Online]. Available: http://www.sciencedirect.com/science/article/B6VGJ-4DCMC0G-1/2/cf49addcd55de34f935d6106c46dfd68

[8] G. Trajcevski, H. Ding, P. Scheuermann, R. Tamassia, and D. Vaccaro, "Dynamics-aware similarity of moving objects trajectories," in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. New York, NY, USA: ACM, 2007, pp. 1–8.

[9] D. Chudova, S. Gaffney, E. Mjolsness, and P. Smyth, "Translation-invariant mixture models for curve clustering," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2003, pp. 79–88.

[10] A. Daniel and S. Thad, "Using gps to learn significant locations and predict movement across multiple users," in *Personal and Ubiquitous Computing*, vol. 7. Springer, 2003, pp. 275–286.

[11] Alejandro Dizan Vasquez Govea, *Incremental Learning for Motion Prediction of Pedestrians and Vehicles*. Springer-Verlag, 2010.

[12] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 25–33.

[13] A. Asahara, K. Maruyama, and R. Shibasaki, "A mixed autoregressive hidden-markov-chain model applied to people's movements," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 414–417. [Online]. Available: http://doi.acm.org/10.1145/2424321.2424378

[14] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986.

[15] J. S. Simonoff, *Smoothing methods in statistics*. Springer, 1996.

[16] Cabinet Office, Government of Japan, "Resilient ITS," http://mnj.gov-online.go.jp/its.html.

[17] Hitachi Information and Telecomunication Engineering, Ltd. Inc., "LaserRadarvisionII (in Japanese)," http://www.hitachi-ite.co.jp/products/lr/.

[18] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser-range scanners," *IEEE Transactions on Man and Cybernetics Systems, Part A: Systems and Humans*, vol. 35, no. 2, pp. 283–291, 2005.

[19] K. Ho-Chul, K. Oje, and L. Ki-Joune, "Spatial and spatiotemporal

analysis of soccer," in *Proceedings of ACM SIGSPATIAL GIS 2011*. New York, NY, USA: ACM.

[20] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, "Interactive humanoid robots for a science museum," in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, ser. HRI '06. New York, NY, USA: ACM, 2006, pp. 305–312. [Online]. Available: http://doi.acm.org/10.1145/1121241.1121293

[21] D. Manandhar, S. Kawaguchi, M. Uchida, M. Ishii, and H. Tomohiro, "IMES for mobile users. social implementation and experiments based on existing cellular phones for seamless positioning," *Proc. of the Int. Symposium on GPS/GNSS 2008*, Nov 2008.

[22] S. Honda, K.-i. Fukui, K. Moriyama, S. Kurihara, and M. Numao, "Extracting human behaviors with infrared sensor network," in *Fourth International Conference on Networked Sensing Systems, 2007. INSS'07*. IEEE, 2007, pp. 122–125.

[23] Y. Teramoto, A. Sato, A. Asahara, and H. Tomita, "Indoor positioning based on radio signal strength distribution modeling using mirror image method," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ser. ISA '11. New York, NY, USA: ACM, 2011, pp. 15–22.

[24] Skyhook, Inc., "Skyhook," http://www.skyhookwireless.com/.

[25] O. Woodman and R. Harle, "Rf-based initialisation for inertial pedestrian tracking," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H. Tokuda, M. Beigl, A. Friday, A. Brush, and Y. Tobe, Eds. Springer Berlin / Heidelberg, 2009, vol. 5538, pp. 238–255.

[26] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. 521–524.

[27] I. Haritaoglu, D. Harwood, and L. Davis, "W^4S: A real-time system for detecting and tracking people in 2 1/2D," in *Computer Vision - ECCV'98*, ser. Lecture Notes in Computer Science, H. Burkhardt and B. Neumann, Eds. Springer Berlin Heidelberg, 1998, vol. 1406, pp. 877–892.

[28] M. Luber, L. Spinello, and K. O. Arras, "People tracking in rgb-d data with on-line boosted target models," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3844–3849.

[29] International Organization for Standardization, "ISO 19141:2008 Geographic information – Schema for moving features," 2008.

[30] Web3D, "X3D," http://www.web3d.org/x3d/, 2008.

[31] Open Geospatial Consortium, "OGC KML," http://www.opengeospatial.org/standards/kml/.

[32] ——, "OGC(R) Moving Features," http://www.opengeospatial.org/standards/movingfeatures.

[33] ——, "Geography Markup Language," http://www.opengeospatial.org/standards/gml/.

[34] Y. Sekimoto, R. Shibasaki, H. Kanasugi, T. Usui, and Y. Shimazaki, "Pflow: Reconstructing people flow recycling large-scale social survey data," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 0027–35, 2011.

[35] geojson.org, "GeoJSON," http://geojson.org/.

[36] OASIS Open, "MQTT Version 3.1.1 OASIS Standard," http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.